

При помощи функции ROW\_NUMBER сгенерируем порядковый номер строки запроса.

```
ПОДСКАЗКА: Возможно, предполагалась ссылка на столбец "magazine.date".
sweet=# select row_number() over (order by magazine) id_product, address from magazine;
id_product | address
-----+-----
          1 | Koltsova
          2 | Revolution
          3 | Pobedi
(3 строки)
```

Функция ABS(n) возвращает абсолютное значение числа n

```
sweet=# select abs(200) x1, abs(-100) x2, abs(0) x3;
x1 | x2 | x3
---+---+---
200 | 100 | 0
(1 строка)
```

Функция CEIL(n) возвращает наибольшее целое, большее или равное переданному в качестве параметра числу n.

```
sweet=# select ceil(100.1) x1, ceil(-100.1) x2, ceil(100.5) x3;
x1 | x2 | x3
---+---+---
101 | -100 | 101
(1 строка)
```

Функция FLOOR(n) возвращает наименьшее целое, меньшее или равное переданному в качестве параметра числу n.

```
sweet=# select floor(150.1) x1, floor(-150.1) x2, floor(150.10) x3;
x1 | x2 | x3
---+---+---
150 | -151 | 150
(1 строка)
```

Функция TRUNC(n, m) возвращает число n, усеченное до m знаков после десятичной точки.

```
sweet=# select trunc(180.1234567, 4) x1, trunc(170.123, 2) x2, trunc(180.23) x3;
x1 | x2 | x3
---+---+---
180.1234 | 170.12 | 180
(1 строка)
```

Функция ROUND(n[, m]) возвращает число n, округленное до m знаков после десятичной точки по правилам математического округления

```
sweet=# select round(100.1234, 2) x1, round(100.555555, 3) x2, round(150.1) x3;
x1 | x2 | x3
---+---+---
100.12 | 100.556 | 150
(1 строка)
```

Функция SIGN(n) определяет знак числа.

```
sweet=# select sign(-5) x1, sign(0) x2, sign(10) x3;
x1 | x2 | x3
---+---+---
-1 | 0 | 1
(1 строка)
```

Функция MOD(n, m) возвращает остаток от деления n на m

```
sweet=# select mod(10,5) x1, mod (11,2) x2;
x1 | x2
---+---
0 | 1
(1 строка)
```

Функция POWER(n, m) возводит число n в степень m  
Функция

```
sweet=# select power(2,6) x1, power(9, 0.5) x2;
x1 | x2
---+---
64 | 3.0000000000000000
(1 строка)
```

Функция SQRT(n) возвращает квадратный корень от числа n.

```
sweet=# select sqrt(64);
sqrt
---
8
(1 строка)
```

Функция EXP(n) возводит e в степень n, а функция LN(n) вычисляет натуральный логарифм от n

```
sweet=# select exp(3) x1, ln(3), ln(exp(3)) x3;
x1 | ln | x3
---+---+---
20.085536923187668 | 1.0986122886681098 | 3
(1 строка)
```

Функция LOG(n, m) производит вычисление логарифма m по основанию n.

```
sweet=# select log(36,6);
log
---
0.5000000000000000
(1 строка)
```

Функции Sin(n), Cos(n), Tan(n), Cot(n) производят вычисление тригонометрических функций.

```
sweet=# select sin(1), cos(1), tan(1), cot(1);
      sin      |      cos      |      tan      |      cot
-----+-----+-----+-----
0.8414709848078965 | 0.5403023058681398 | 1.5574077246549023 | 0.6420926159343306
(1 строка)
```

Функция CONCAT(str1, str2) выполняет склеивание строк str1 и str2

```
sweet=# select concat('hel', 'low');
concat
-----
hellow
(1 строка)
```

Функция LOWER(str) преобразует все символы строки str в строчные

```
sweet=# select lower('HeLlOw');
lower
-----
hellow
(1 строка)
```

Функция UPPER(str) преобразует все символы строки str в прописные

```
sweet=# select upper('HeLlOw');
upper
-----
HELLOW
(1 строка)
```

Функция INITCAP(str) возвращает строку str, в которой первые буквы всех слов преобразованы в прописные.

```
sweet=# select initcap('heLlOw wORLD');
initcap
-----
Hellow World
(1 строка)
```

Функция LTRIM(str, [,set]) удаляет все символы с начала строки до первого символа, которого нет в наборе символов set.

```
sweet=# select ltrim('123789', '123');
ltrim
-----
789
(1 строка)
```

Функция RTRIM(str, [,set]) аналогична, но удаляет символы, начиная от конца строки.

```
sweet=# select rtrim('123789', '987');
rtrim
-----
123
(1 строка)
```

Функция TRANSLATE(str, from\_mask, to\_mask) анализирует строку str и заменяет в ней все символы, встречающиеся в строке from\_mask, на соответствующие символы из to\_mask

```
sweet=# select replace('Hellow world nice evening', 'evening', 'day');
replace
-----
Hellow world nice day
(1 строка)
```

Функция LENGTH(str) возвращает длину строки str в символах.

```
sweet=# select length('world'), length('');
length | length
-----+-----
5 | 0
(1 строка)
```

Функция ASCII(str) возвращает ASCII-код первого символа строки str в случае применения кодировок ASCII и UTF-8.

```
sweet=# select ascii('world');
ascii
-----
119
(1 строка)
```

Функция CHR(n) возвращает символ по его коду

```
sweet=# select chr(70);
chr
-----
F
(1 строка)
```

Функция NOW() возвращает текущую дату и время по часам сервера

```
sweet=# select now();
now
-----
2023-05-20 03:04:19.728186+03
(1 строка)
```

Функция JUSTIFY\_INTERVAL(interval) преобразует интервал, указанный в виде строки в соответствующее значение

```
sweet=# select justify_interval('4 days 4 hours 4 minute 44 second');
justify_interval
-----
4 days 04:04:44
(1 строка)
```

Функция DATE\_TRUNC(timestamp) используется для обрезки даты или интервала (DATE\_TRUNC(interval)) до определенной точности

```
sweet=# select date_trunc('hour', now());
date_trunc
-----
2023-05-20 03:00:00+03
(1 строка)
```

Функция AGE([end\_date, ]start\_date) возвращает разницу между датами, обозначенными как end\_date и start\_date.

```
sweet=# select age(now(), '2020-9-1 9:45:00');
age
-----
2 years 8 mons 18 days 17:22:31.125046
(1 строка)
```

Функция EXTRACT(field FROM timestamp) извлекает элемент даты field из значения типа timestamp

```
sweet=# select now(),extract(minute from now());
now | extract
-----+-----
2023-05-20 03:08:36.970268+03 | 8
(1 строка)
```

Функция TO\_DATE(str, mask) преобразует строку str в дату.

```
sweet=# select to_date('8 feb 2002', 'dd mon yyyy');
to_date
-----
2002-02-08
(1 строка)
```

Функция TO\_CHAR(date, mask) преобразует дату date в символьную строку в соответствии с заданной маской

```
sweet=# select to_char(now(), 'yy.mm.dd;')
to_char
-----
23.05.20;
(1 строка)
```